

Fourier Accelerated Conjugate Gradient Lattice Gauge Fixing

R.J. Hudspith¹

(The RBC and UKQCD Collaborations)

¹*Department of Physics and Astronomy, York University, Toronto, Ontario, M3J 1P3, Canada*

We provide details of the first implementation of a non-linear conjugate gradient method for Landau and Coulomb gauge fixing with Fourier acceleration. We find clear improvement over the Fourier accelerated steepest descent method, with the average time taken for the algorithm to converge to a fixed, high accuracy, being reduced by a factor of 2 to 4.

INTRODUCTION

Conjugate gradient (CG) methods (first used to solve linear equations [1] and later generalised for non-linear, non-quadratic functions [2, 3]) are a technique to solve unconstrained local minimisation problems. Numerically these methods can be implemented cheaply because they are iterative and converge in a finite number of steps. They are also often considered computationally faster than the steepest descent method [4]. We illustrate how fixing to the smooth, Landau and Coulomb gauges in the context of lattice field theory can be achieved by using the method of conjugate gradients.

Fixing the gauge is a prescription for removing redundant degrees of freedom of the gauge field in a continuum quantum field theory. Common choices are the Landau $\partial_\mu A_\mu(x) = 0$ and Coulomb $\partial_i A_i(x) = 0$ gauges (where Greek indices run over all dimensions and Roman over the spatial and the A 's are the gauge fields of our theory). Fixing the gauge (while not necessary for many lattice measurements) is often required for the direct matching of lattice simulations to continuum perturbation theory.

Measurements of lattice Green's functions in strongly coupled, N_d -dimensional, $SU(N_c)$ theories have to be performed with a fixed gauge and are often computed in Landau gauge. Landau gauge Green's functions are vital for the non-perturbative renormalisation of important physical quantities such as the QCD Kaon bag parameter B_K [5, 6] and can also be used for the measurement of the QCD strong coupling α_s [7]. Coulomb gauge fixing is more generally applicable to lattice theorists; it is used in methods such as gauge-fixed wall source quark correlators (also often used in the calculation of B_K), or for computing the static quark potential [8]. Having a fast routine to fix the gauge allows for faster measurements of critical physical quantities.

Lattice "links" transcribe the gauge fields via the matrix exponential (for lattice site " x ", with lattice spacing a and bare coupling g_0),

$$U_\mu \left(x + a \frac{\hat{\mu}}{2} \right) = e^{i a g_0 A_\mu \left(x + a \frac{\hat{\mu}}{2} \right)}. \quad (1)$$

The gauge fields are obtained by the logarithm of the links. A common approximation [9] to the logarithm of

the map $U = e^{iA}$, $U \in SU(N_c)$ is what we call the "Hermitian projection" (where $I_{N_c \times N_c}$ is the identity matrix),

$$A = \frac{1}{2i} \left\{ [U - U^\dagger] - \frac{1}{N_c} \text{Tr} [U - U^\dagger] \cdot I_{N_c \times N_c} \right\}. \quad (2)$$

This definition is not unique, being correct up to terms of $O(A^3)$. Exact logarithm techniques are possible [10, 11] but will not be discussed here as they are numerically costly to implement and less commonly used in practice.

GAUGE FIXING ON A LATTICE

We now discuss the case of lattice Landau gauge fixing, as the extension to Coulomb gauge should be simple. After we introduce the conjugate gradient procedure we then discuss our implementation for fixing to Coulomb gauge.

There are two common types of gauge fixing routines, the Los Alamos [9] and the Cornell [12], both of which use the method of steepest descent to minimise the functional (where V is the lattice volume),

$$F(U) = \frac{1}{N_d N_c V} \sum_{x, \mu} \text{Tr} \left[\left(a g_0 A_\mu \left(x + a \frac{\hat{\mu}}{2} \right) \right)^2 \right]. \quad (3)$$

We focus on the Cornell method as it can be Fourier accelerated.

For the Hermitian projection definition (Eq.2) of the gauge fields, the following approximation of the functional can be used,

$$F(U) \approx 1 - \frac{1}{N_d N_c V} \sum_{x, \mu} \Re \left(\text{Tr} \left[U_\mu \left(x + a \frac{\hat{\mu}}{2} \right) \right] \right). \quad (4)$$

In general, the method of steepest descent is a technique to find a local minimum of a function. Considering the n^{th} iteration of such a method, the update,

$$x_{n+1} = x_n - \alpha f'(x_n), \quad (5)$$

will step toward a local minimum of the function $f(x)$ provided the parameter is small, $0 < \alpha < 1$.

In direct analogy to the general procedure of Eq.5 we first approximate the derivative of a gauge field by,

$$a\Delta_\mu A_\mu(x) = \sum_\mu \left(A_\mu \left(x + a\frac{\hat{\mu}}{2} \right) - A_\mu \left(x - a\frac{\hat{\mu}}{2} \right) \right). \quad (6)$$

The n^{th} iteration of the lattice steepest descent Landau gauge fixing procedure updates the links via the gauge transformation,

$$g(x) = e^{-i\alpha a\Delta_\mu a g_0 A_\mu^{(n)}(x)},$$

$$U_\mu^{(n+1)} \left(x + a\frac{\hat{\mu}}{2} \right) = g(x) U_\mu^{(n)} \left(x + a\frac{\hat{\mu}}{2} \right) g(x + a\hat{\mu})^\dagger. \quad (7)$$

This procedure successively minimises $a\Delta_\mu a g_0 A_\mu^{(n)}(x)$, whilst retaining the gauge invariance of the action. The parameter α is again a small tuning parameter, which could be tuned at each step but is best fixed to a near-optimal constant value. We found setting α to 0.08 and 0.1 for Landau and Coulomb gauge respectively to be best for the ensembles considered in this work.

The exponential in Eq.7 could be computed exactly using the technique of [13], but expansion to the term linear in α and reunitarisation is sufficient and numerically faster.

It is common to stop the gauge fixing routine once the quantity,

$$\Theta^{(n)} = \frac{1}{N_c V} \sum_x \text{Tr} \left[\left(a\Delta_\mu a g_0 A_\mu^{(n)}(x) \right)^2 \right], \quad (8)$$

reaches some small value (often $\Theta^{(n)} \approx 10^{-14}$).

Fourier acceleration

The steepest descent method of fixing to Landau gauge as outlined above was shown in [12] to suffer from critical slowing down, meaning that the number of iterations required to converge to a fixed accuracy grows drastically with the volume of the problem. Their method to ameliorate this was to apply a re-scaling in momentum space of the eigenvalues of the (Abelian) Laplacian Δ^2 ,

$$g(x) = e^{-i\alpha \tilde{F} \frac{p_{\text{Max}}^2}{V p^2} F a\Delta_\mu a g_0 A_\mu^{(n)}(x)}, \quad (9)$$

where F and \tilde{F} are forward and backward fast Fourier transforms (FFTs) respectively, and the factor of $\frac{1}{V}$ is for the FFT normalisation. In practice, it is best if the quantities $\frac{p_{\text{Max}}^2}{V p^2}$ are precomputed as a look up table. The discrete momenta p^2 have the usual lattice definition,

$$p^2 = 2 \left(N_d - \sum_\mu \cos \left(\frac{2\pi n_\mu}{L_\mu} \right) \right), \quad (10)$$

with Fourier modes $n_\mu = \left(\frac{-L_\mu}{2}, \dots, -1, 0, 1, \dots, \frac{L_\mu}{2} - 1 \right)$. L_μ is the length of the lattice in the μ direction. Special care should be taken at the zero mode, where we set the value of p^2 to 1 [12].

In our implementation of Fourier acceleration, the shared-memory parallel version of the library FFTW [14] was used. The Fourier accelerated steepest descent method will be denoted as FASD later on in this paper.

THE CONJUGATE GRADIENT METHOD

An outline of the general approach for the non-linear (Polyak-Ribieré [3])[18] conjugate gradient method is shown in Alg.1.

Algorithm 1 General non-linear CG

```

Compute the gradient direction  $f'(x_0)$ 
Perform a line search for  $\alpha_0$  s.t  $\min(f(x_0 - \alpha_0 f'(x_0)))$ 
Perform the update  $x_1 = x_0 - \alpha_0 f'(x_0)$ 
Set  $s_0 = -f'(x_0)$ 
 $n = 1$ 
while  $|f'(x_n)|^2 > \text{Tolerance}$  do
  Compute the gradient  $f'(x_n)$ 
  Compute  $\beta_n = \max \left[ 0, \frac{f'(x_n)^T (f'(x_n) - f'(x_{n-1}))}{f'(x_{n-1})^T f'(x_{n-1})} \right]$ 
  Compute conjugate direction  $s_n = -f'(x_n) + \beta_n s_{n-1}$ 
  Perform a line search for  $\alpha_n$  s.t  $\min(f(x_n + \alpha_n s_n))$ 
  Update  $x_{n+1} = x_n + \alpha_n s_n$ 
   $n = n + 1$ 
end while

```

The translation of this approach to lattice Landau gauge fixing follows almost directly, and we call it the Fourier Accelerated Conjugate Gradient (FACG) method, and outline it in Alg.2. This algorithm should not be confused with the CGFA algorithm of [15], which uses the CG algorithm to invert the Laplacian instead of performing FFTs.

The approach begins with an FASD step as in Eq.9, storing the result of the Fourier accelerated derivative in the conjugate direction $s_n(x)$. Once the algorithm has reached a sufficient minimum we are finished, otherwise we repeat the procedure generating conjugate directions weighted by the factor β_n .

We choose to use a line search to approximately determine the optimal tuning parameter α_n at each iteration. To do so we evaluate the gauge fixing functional (Eq.4) for possible fixed probe values of the parameter α , which we call α' and create a cubic spline interpolation of the result. We then solve for the exact minimum of the cubic spline.

The evaluation of each probe α' is the most expensive aspect of this approach as they each require both the exponentiation of the derivatives into the gauge transformation matrices, and a subsequent gauge transformation over the whole lattice. Therefore, performing the bare

minimum number of evaluations in this step is key to a fast implementation of this procedure.

For both Landau and Coulomb gauge, the probes $\alpha' \in (0.0, 0.15, 0.3)$ were used. The point at $\alpha' = 0$ is the cheapest probe to evaluate because no extra exponentiations and gauge transformations are required; it should always be used.

Algorithm 2 Landau gauge FACG

```

 $\Gamma_0(x) \leftarrow \tilde{F} \frac{p_{\text{Max}}^2}{V p^2} F a \Delta_\mu a g_0 A_\mu^{(0)}(x)$ 
 $g(x) \leftarrow e^{-i\alpha_0 \Gamma_0(x)}$ 
 $U_\mu^{(1)}(x + a \frac{\hat{\mu}}{2}) \leftarrow g(x) U_\mu^{(0)}(x + a \frac{\hat{\mu}}{2}) g(x + a \hat{\mu})^\dagger$ 
 $s_0(x) \leftarrow \Gamma_0(x)$ 
 $n = 1$ 
while  $\Theta^{(n)} > \text{Tolerance}$  do
   $\Gamma_n(x) \leftarrow \tilde{F} \frac{p_{\text{Max}}^2}{V p^2} F a \Delta_\mu a g_0 A_\mu^{(n)}(x)$ 
   $\beta_n = \max \left[ 0, \frac{\sum_x \text{Tr}[\Gamma_n(x)(\Gamma_n(x) - \Gamma_{n-1}(x))]}{\sum_x \text{Tr}[\Gamma_{n-1}(x)^2]} \right]$ 
   $s_n(x) \leftarrow \Gamma_n(x) + \beta_n s_{n-1}(x)$ 
   $\alpha_n = \min(F(U); \alpha' s_n)$ 
   $g(x) \leftarrow e^{-i\alpha_n s_n(x)}$ 
   $U_\mu^{(n+1)}(x + a \frac{\hat{\mu}}{2}) \leftarrow g(x) U_\mu^{(n)}(x + a \frac{\hat{\mu}}{2}) g(x + a \hat{\mu})^\dagger$ 
   $n = n + 1$ 
end while

```

The function $\min(F(U); \alpha' s_n)$ finds the value of α_n such that

$$1 - \frac{1}{N_d N_c V} \sum_{x, \mu} \Re \left(\text{Tr} \left[e^{-i\alpha_n s_n(x)} U_\mu^{(n)} \left(x + a \frac{\hat{\mu}}{2} \right) e^{i\alpha_n s_n(x + a \hat{\mu})} \right] \right),$$

is approximately best minimised.

As an optimisation, due to the Hermiticity of the gauge fields and hence their derivatives, we only need to Fourier transform the upper or lower triangular part of the matrix $a \Delta_\mu a g_0 A_\mu(x)$. The rest of the matrix can be reconstructed by Hermiticity and the final element of the matrix by tracelessness. Not only does this reduce the number of FFTs the routine performs but it also reduces the memory required to store these matrices.

A negative aspect of this method is that extra storage of the matrices $\Gamma_n(x)$ and $s_n(x)$ has to be made available, although these can be stored in the shortened form as mentioned above.

As the gauge fixing progresses and $\Theta^{(n)}$ approaches zero, the functional (Eq.4) flattens so that within finite numerical precision the minimum is indistinguishable for any probe α' . Once we reach this accuracy (approximately the machine epsilon of the storage precision of our fields) we switch to the Fletcher-Reeves [2] definition of $\beta_n = \frac{\sum_x \text{Tr}[\Gamma_n^2]}{\sum_x \text{Tr}[\Gamma_{n-1}^2]}$ and fixed α_n (i.e. we turn off the line search altogether).

COULOMB GAUGE

Fixing to Coulomb gauge on the lattice amounts to the minimisation of the functional,

$$F(U) = \frac{1}{(N_d - 1) N_c V} \sum_{x, i} \text{Tr} \left[\left(a g_0 A_i \left(x + a \frac{\hat{i}}{2} \right) \right)^2 \right],$$

where the Roman index “ i ” runs over the spatial indices only. The functional is time-slice independent; by exploiting this a time-slice by time-slice scheme can be a very effective method in fixing to Coulomb gauge.

It is observed that each time-slice takes a greatly different number of iterations to converge compared to one another [16]. If we used a lattice-wide technique, we could spend a lot of time waiting for the slowest converging time-slices as well as requiring N_d -dimensional, volume-wide FFTs. This advocates an approach which treats each time-slice separately. We will detail our implementation of a time-slice by time-slice FASD procedure as extension of this to an FACG procedure is straightforward.

We define the lattice gauge field derivative for fields restricted to a time-slice “ t ”,

$$a \Delta_i A_i(x, t) = \sum_i \left(A_i \left(x + a \frac{\hat{i}}{2}, t \right) - A_i \left(x - a \frac{\hat{i}}{2}, t \right) \right).$$

The n^{th} iteration of the FASD method for Coulomb gauge fixing requires the computation of the gauge transformation matrices ($g^{(0)}(x, t) = I_{N_c \times N_c}$),

$$g^{(n+1)}(x, t) = e^{-i\alpha \tilde{F} \frac{p_{\text{Max}}^2}{V_{N_d-1} p^2} F a \Delta_i a g_0 A_i^{(n)}(x, t)} g^{(n)}(x, t). \quad (11)$$

The Fourier transforms and momenta are defined in the $N_d - 1$ dimensional (spatial) subspace whose volume is denoted V_{N_d-1} , with $p_{\text{Max}}^2 = 4(N_d - 1)$.

As expressed in Eq.11, we store the gauge transformation matrices as an accumulated product of the previous iterations. Therefore, each computation of the derivatives requires a local gauge transformation of the original links, and we do not overwrite the link matrices on that time-slice until the routine has converged. The Hermitian projection definition of our fields (Eq.2) then reads (where the subscript Trf refers to the trace subtraction),

$$a g_0 A_i^{(n)} \left(x + a \frac{\hat{i}}{2}, t \right) = \frac{1}{2i} \left[g^{(n)}(x, t) U_i \left(x + a \frac{\hat{i}}{2}, t \right) g^{(n)}(x + a \hat{i}, t)^\dagger - g^{(n)}(x + a \hat{i}, t) U_i \left(x + a \frac{\hat{i}}{2}, t \right)^\dagger g^{(n)}(x, t) \right]_{\text{Trf}}. \quad (12)$$

We use the criterion (where the sum is over the fields on a time-slice),

$$\theta^{(n)}(t) = \frac{1}{N_c V_{N_d-1}} \sum_x \text{Tr} \left[\left(a \Delta_i a g_0 A_i^{(n)}(x, t) \right)^2 \right],$$

to estimate our convergence.

Once we have converged the gauge transformation matrices for a time-slice $g(x, t)$ and the ones above it $g(x, t + a)$, we can overwrite the link matrices by gauge transformation,

$$\begin{aligned} U_i \left(x + a \frac{\hat{i}}{2}, t \right) &\leftarrow g(x, t) U_i \left(x + a \frac{\hat{i}}{2}, t \right) g(x + a \hat{i}, t)^\dagger, \\ U_t(x, t) &\leftarrow g(x, t) U_t(x, t) g(x, t + a)^\dagger. \end{aligned} \quad (13)$$

This ensures the gauge invariance of the action under this procedure.

The algorithm for our procedure is presented in Alg.3, illustrating the use of temporary, time-slice wide, gauge transformation matrices (called p, q and r) as a computer memory saving practice. It should be noted that the spatial gauge transformations for the $A_i^{(n)}$'s (Eq.12) are now performed in terms of the p 's, q 's and r 's within the respective “while” loops.

Algorithm 3 Slice-by-slice Coulomb gauge FASD

```

 $n = 0, p(x) \leftarrow I_{N_c \times N_c}$ 
while  $\theta^{(n)}(0) > \text{Tolerance}$  do
     $p(x) \leftarrow e^{-i\alpha \tilde{F} \frac{p_{\text{Max}}^2}{V_{N_d-1} p^2} F a \Delta_i a g_0 A_i^{(n)}(x, 0)} p(x)$ 
     $n = n + 1$ 
end while
 $n = 0, q(x) \leftarrow I_{N_c \times N_c}$ 
while  $\theta^{(n)}(1) > \text{Tolerance}$  do
     $q(x) \leftarrow e^{-i\alpha \tilde{F} \frac{p_{\text{Max}}^2}{V_{N_d-1} p^2} F a \Delta_i a g_0 A_i^{(n)}(x, 1)} q(x)$ 
     $n = n + 1$ 
end while
 $U_i \left( x + a \frac{\hat{i}}{2}, 0 \right) \leftarrow p(x) U_i \left( x + a \frac{\hat{i}}{2}, 0 \right) p(x + a \hat{i})^\dagger$ 
 $U_t(x, 0) \leftarrow p(x) U_t(x, 0) q(x)^\dagger$ 
for  $t = 2 \rightarrow L_t - a$  do
     $n = 0, r(x) \leftarrow I_{N_c \times N_c}$ 
    while  $\theta^{(n)}(t) > \text{Tolerance}$  do
         $r(x) \leftarrow e^{-i\alpha \tilde{F} \frac{p_{\text{Max}}^2}{V_{N_d-1} p^2} F a \Delta_i a g_0 A_i^{(n)}(x, t)} r(x)$ 
         $n = n + 1$ 
    end while
     $U_i \left( x + a \frac{\hat{i}}{2}, t - a \right) \leftarrow q(x) U_i \left( x + a \frac{\hat{i}}{2}, t - a \right) q(x + a \hat{i})^\dagger$ 
     $U_t(x, t - a) \leftarrow q(x) U_t(x, t - a) r(x)^\dagger$ 
     $q(x) \leftarrow r(x)$ 
end for
 $U_i \left( x + a \frac{\hat{i}}{2}, L_t - a \right) \leftarrow q(x) U_i \left( x + a \frac{\hat{i}}{2}, L_t - a \right) q(x + a \hat{i})^\dagger$ 
 $U_t(x, L_t - a) \leftarrow q(x) U_t(x, L_t - a) p(x)^\dagger$ 

```

Occasionally, the algorithm on a time-slice converges slowly. This is due to the initial topography of the fields,

and is remedied by performing a random gauge transformation on the initial link matrices if n is larger than some predetermined number and $\theta^{(n)}(t)$ is not close to the desired accuracy.

Storing our gauge transformation matrices as an accumulated product can lead to small round off errors, which we compensate for by reunitarisation before applying the gauge transformation (Eq.13) to the links.

The extension of the time-slice by time-slice FASD routine outlined in Alg.3 to a FACG routine requires the replacement of each while loop that sets the spatial gauge transformation matrices p, q and r with a spatial-polarisation variant of Alg.2. As in the FASD case, we compute the accumulated product of the gauge transformation matrices instead of overwriting the links at each step for our Coulomb gauge FACG method.

RESULTS

Fig.1 illustrates a measure of the effective speed up for our conjugate gradient approach by showing the average time taken to achieve some common accuracy using the FASD algorithm at near-optimal fixed step α , divided by the time taken to achieve the same accuracy for the FACG algorithm for both Landau and Coulomb gauge. The results are generated from the same (SU(3), $N_d = 4$) gauge configuration, randomly gauge transformed 25 times, for 3 different Iwasaki gauge action DWF ensembles with the same lattice spacing of $a^{-1} \approx 1.75$ GeV [17] and different physical volumes.

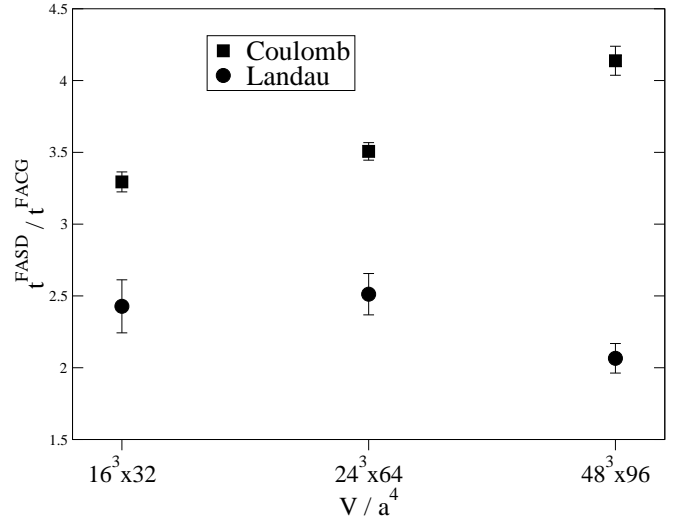


FIG. 1: The improvement factor $\frac{t_{\text{FASD}}}{t_{\text{FACG}}}$ defined by the ratio of the time taken to achieve a gauge fixing accuracy of $\Theta^{(n)} < 10^{-14}$ for both the Landau and Coulomb FASD to FACG routines.

The time taken to achieve a fixed accuracy is a more

representative measurement of the real-world application of this procedure than say the time taken for the algorithm to perform a set number of iterations. This is because the number of iterations required to achieve fixed accuracy grows with volume (a sure sign that the Fourier acceleration is not removing all critical slowing down). Although one iteration should either scale like $O(V \log(V))$ or $O(V)$ depending on whether the cost in performing the FFT is dominant or not, the total time taken to achieve a fixed accuracy is measured to grow like V^n where $1 < n < 2$.

Volume	Landau		Coulomb	
	FASD (s)	FACG (s)	FASD (s)	FACG (s)
$16^3 \times 32$	25(1)	10.5(5)	27.4(6)	8.3(1)
$24^3 \times 64$	306(18)	122(7)	194(3)	55.4(4)
$48^3 \times 96$	12995(857)	6292(404)	10882(180)	2629(39)

TABLE I: The time in seconds taken to achieve an accuracy of $\Theta^{(n)} < 10^{-14}$ for 25 randomly transformed copies of the same, well-thermalised configuration. Each configuration has the same lattice spacing. This measurement was performed using four 8-core AMD Opteron 6128 processors. All operations were performed in double precision.

From Fig.1 we see that the algorithm gives roughly a $\approx 3.5\times$ speed up over our FASD implementation for Coulomb gauge and better performance going to larger volumes. For Landau gauge we achieve a more modest $\approx 2.5\times$, with a slight drop in performance as we go to large volumes. Tab.I gives the raw data for Fig.1, and we see that even for large lattices the problem is particularly tractable even on small clusters of CPUs.

CONCLUSIONS

Fixing the gauge is often necessary for lattice analyses. It should not, however, be a computational bottleneck as fast algorithms with good scaling properties exist. Choosing the best algorithm should take precedence over the use of expensive hardware and aggressive optimisations when implementing a gauge fixing procedure.

We have introduced a new method for fixing to Landau and Coulomb gauge that is shown to be faster by a factor of more than 2 to 4 times respectively over the commonly used Fourier accelerated steepest descent method, albeit for a very moderate increase in required computer memory. We conclude that the upgrade from a FASD to a FACG routine is straightforward and worthwhile.

Implementations of the FASD and FACG algorithm for both Landau and Coulomb gauge are available as part of the package GLU, which is an open source shared-memory parallel gluonic observable library and is available for download at <https://github.com/RJHudspith/GLU>.

ACKNOWLEDGEMENTS

The results produced in this work were generated on Columbia University's cluster "CUTH". RJH would like to thank M.Wurtz, N. Garron and R. Lewis for providing useful comments about this manuscript. RJH is supported by the Natural Sciences and Engineering Research Council of Canada.

-
- [1] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems.*, *J. Res. Natl. Bur. Stand.* **49** (1952) 409–436.
 - [2] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, *The Computer Journal* **7** (1964), no. 2 149–154.
 - [3] B. Polyak, *The conjugate gradient method in extremal problems*, *{USSR} Computational Mathematics and Mathematical Physics* **9** (1969), no. 4 94112.
 - [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.
 - [5] Y. Aoki, R. Arthur, T. Blum, P. Boyle, D. Brommel, et al., *Continuum Limit of B_K from 2+1 Flavor Domain Wall QCD*, *Phys.Rev.* **D84** (2011) 014503, [[arXiv:1012.4178](#)].
 - [6] **RBC Collaboration, UKQCD Collaboration** Collaboration, P. Boyle, N. Garron, and R. Hudspith, *Neutral kaon mixing beyond the standard model with $n_f = 2 + 1$ chiral fermions*, *Phys.Rev.* **D86** (2012) 054028, [[arXiv:1206.5737](#)].
 - [7] **ETM Collaboration** Collaboration, B. Blossier et al., *High statistics determination of the strong coupling constant in Taylor scheme and its OPE Wilson coefficient from lattice QCD with a dynamical charm*, *Phys.Rev.* **D89** (2014) 014507, [[arXiv:1310.3763](#)].
 - [8] O. Kaczmarek, F. Karsch, P. Petreczky, and F. Zantow, *Heavy quark anti-quark free energy and the renormalized Polyakov loop*, *Phys.Lett.* **B543** (2002) 41–47, [[hep-lat/0207002](#)].
 - [9] J. Mandula and M. Ogilvie, *The gluon is massive: A lattice calculation of the gluon propagator in the landau gauge*, *Physics Letters B* **185** (1987), no. 12 127 – 132.
 - [10] S. Durr, *Logarithmic link smearing for full QCD*, *Comput.Phys.Commun.* **180** (2009) 1338–1357, [[arXiv:0709.4110](#)].
 - [11] E.-M. Ilgenfritz, C. Menz, M. Muller-Preussker, A. Schiller, and A. Sternbeck, *$SU(3)$ Landau gauge gluon and ghost propagators using the logarithmic lattice gluon field definition*, *Phys.Rev.* **D83** (2011) 054506, [[arXiv:1010.5120](#)].
 - [12] C. T. H. Davies, G. G. Batrouni, G. R. Katz, A. S. Kronfeld, G. P. Lepage, K. G. Wilson, P. Rossi, and B. Svetitsky, *Fourier acceleration in lattice gauge theories. i. landau gauge fixing*, *Phys. Rev. D* **37** (Mar, 1988) 1581–1588.
 - [13] C. Morningstar and M. Peardon, *Analytic smearing of $su(3)$ link variables in lattice qcd*, *Phys. Rev. D* **69** (Mar, 2004) 054501.

- [14] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, *Proceedings of the IEEE* **93** (2005), no. 2 216–231. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- [15] A. Cucchieri and T. Mendes, *A Multigrid implementation of the Fourier acceleration method for Landau gauge fixing*, *Phys.Rev.* **D57** (1998) 3822–3826, [[hep-lat/9711047](#)].
- [16] Y. Nakagawa, A. Voigt, E.-M. Ilgenfritz, M. Muller-Preussker, A. Nakamura, et al., *Coulomb-gauge ghost and gluon propagators in $SU(3)$ lattice Yang-Mills theory*, *Phys.Rev.* **D79** (2009) 114504, [[arXiv:0902.4321](#)].
- [17] **RBC Collaboration, UKQCD Collaboration** Collaboration, R. Arthur et al., *Domain Wall QCD with Near-Physical Pions*, [arXiv:1208.4412](#).
- [18] We find that the Polyak-Ribieré definition of β_n reduces the iteration count in comparison to the Fletcher-Reeves.